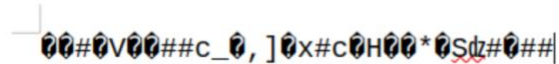


Lab 05 Template – Ethan Roepke

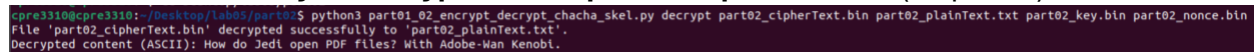
Part 01

1. **Open the part01_cipherText.bin file and include a screenshot of the result in your lab report**(15 points)



Part 02

2. **Include a screenshot in your lab report of the resulting ascii text printed on your screen when you decrypted from part02_cipherText.bin**(15 points)

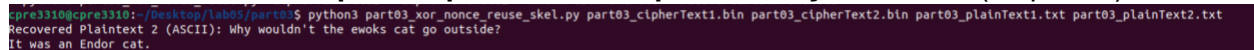


3. **Upload your working part01_02_skel.py****part01_02_encrypt_decrypt_chacha_skel.py code as a separate file**(20 points)

UPLOADED

Part 03

4. **Include a screenshot in your lab report of the resulting ascii text printed on your screen which is part03_plainText2.txt printed on your screen**(15 points)



5. Questions related to reusing nonce, but not the key
 - a. **Would you generate a unique keystream if the nonce was reused, but the key changed?**(5 points)

No I would not generate a unique keystream because when the key has changed, the nonce will be altered that will lead to a different keystream. If we reused the nonce and the key then yes I would generate a unique keystream to prevent vulnerabilities.
 - b. **Would you really want to implement ChaCha20 reusing the nonce? Why or why not?**(10 points)

No, reusing the nonce with the same key would generate the same keystream for multiple messages. Chacha20 is designed to avoid vulnerabilities by reusing the key and nonce for multiple messages. It is designed to require a unique nonce for every encryption but reusing the nonce negates the use of chacha20.

6. **Upload your working part03_xor_reuse_nonce_skel.py code as a separate file**(20 points)

UPLOADED